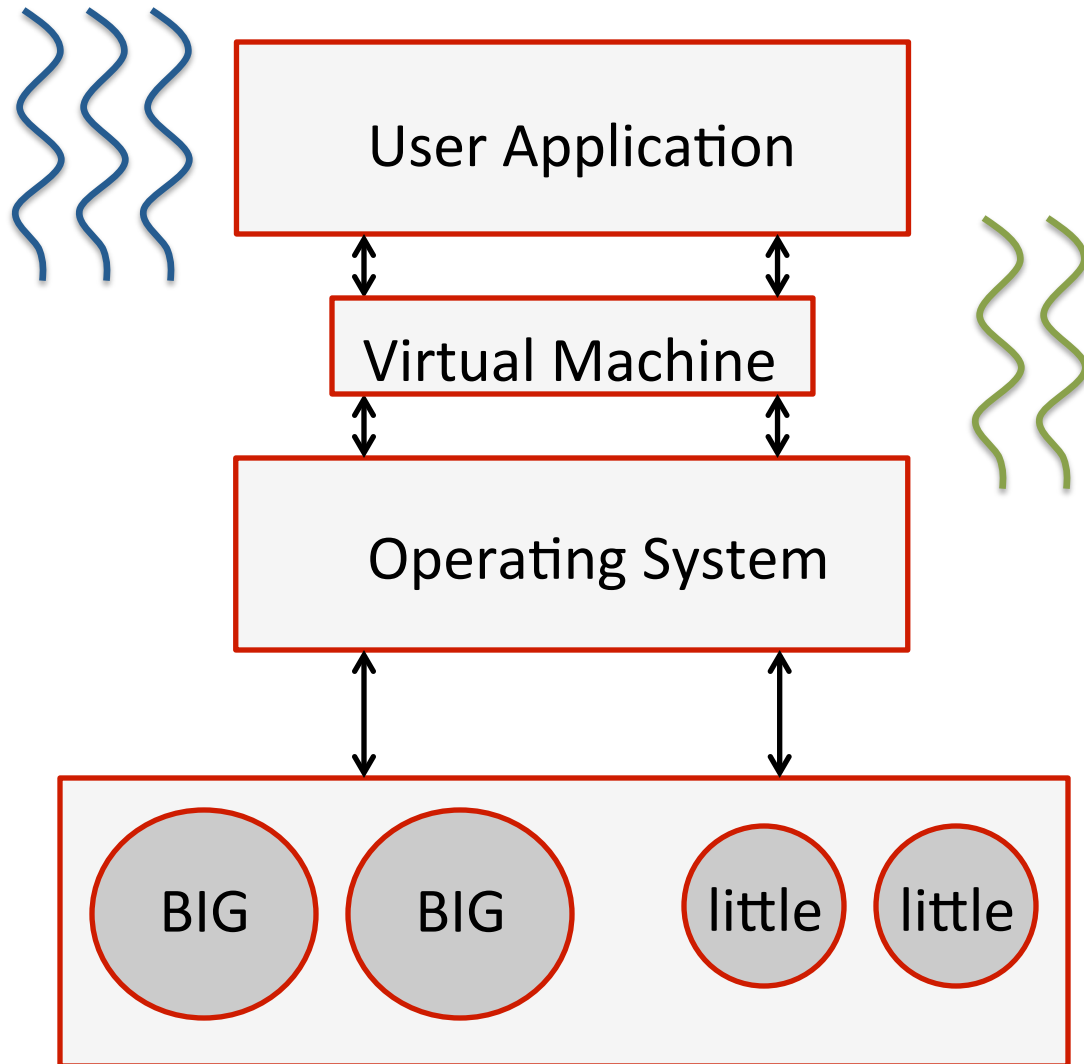


Boosting the Priority of Garbage: Scheduling Collection on Heterogeneous Multicore Processors [TACO 2016]

S. Akram, J.B. Sartor, K. Van Craeynest,
W. Heirman, and L. Eeckhout

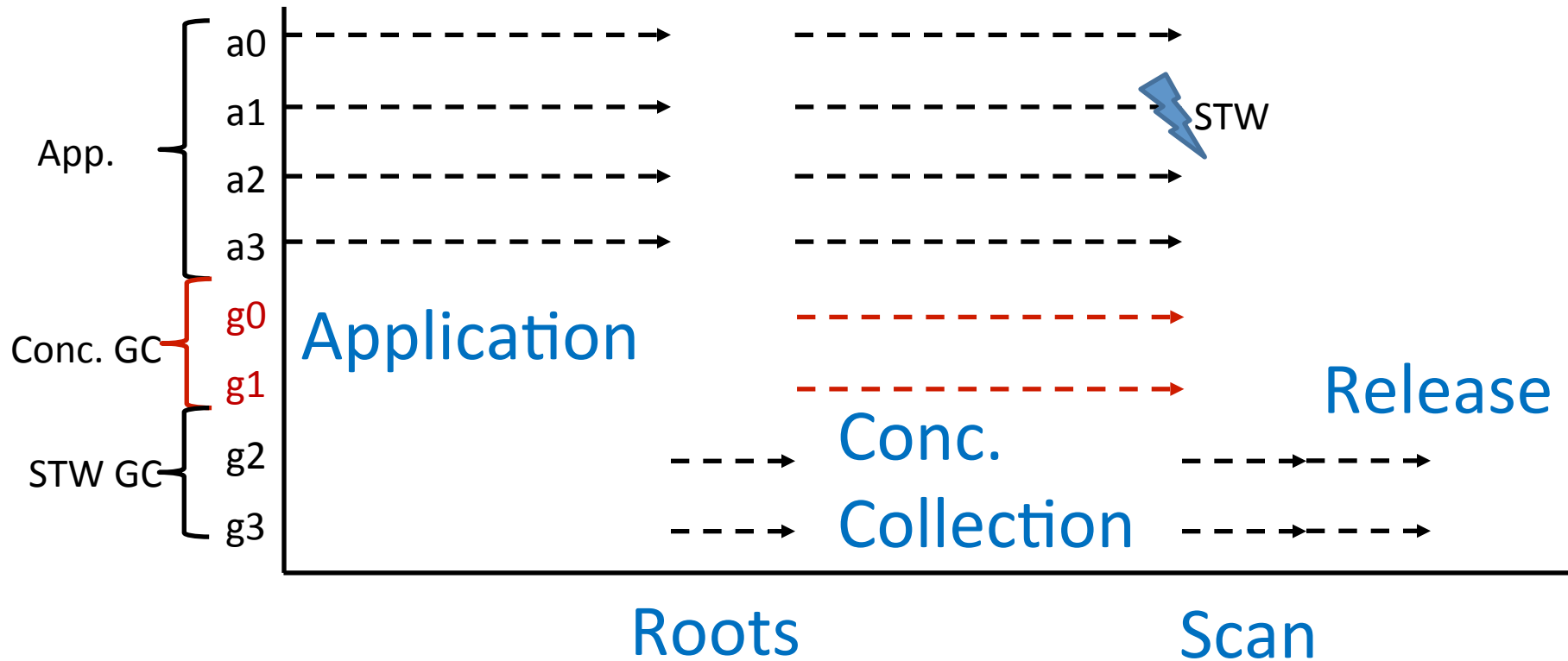


Previous Scheduling Work

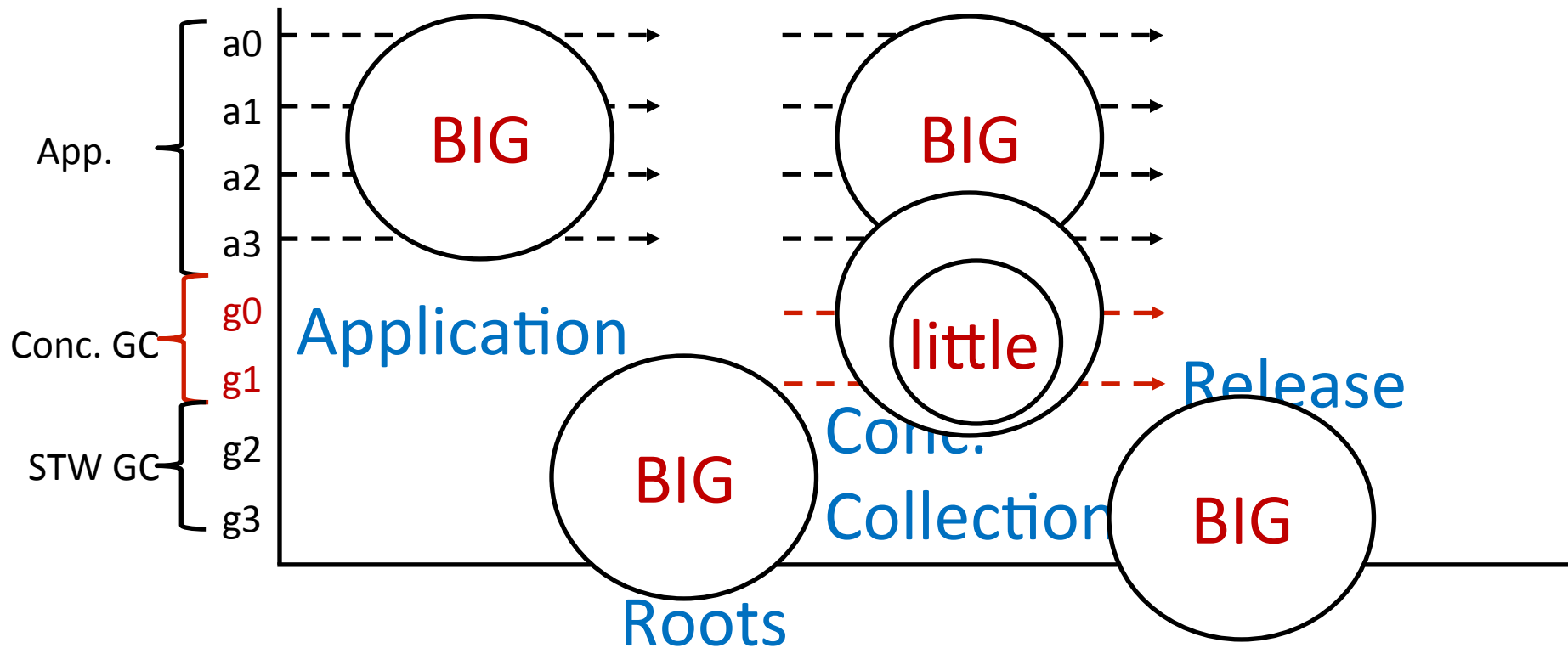


T. Cao, S.M. Blackburn, T. Gao and K.S. McKinley. The Yin and Yang of Power and Performance for Asymmetric Hardware and Managed Software. ISCA 2012

If Concurrent GC Cannot Keep Up

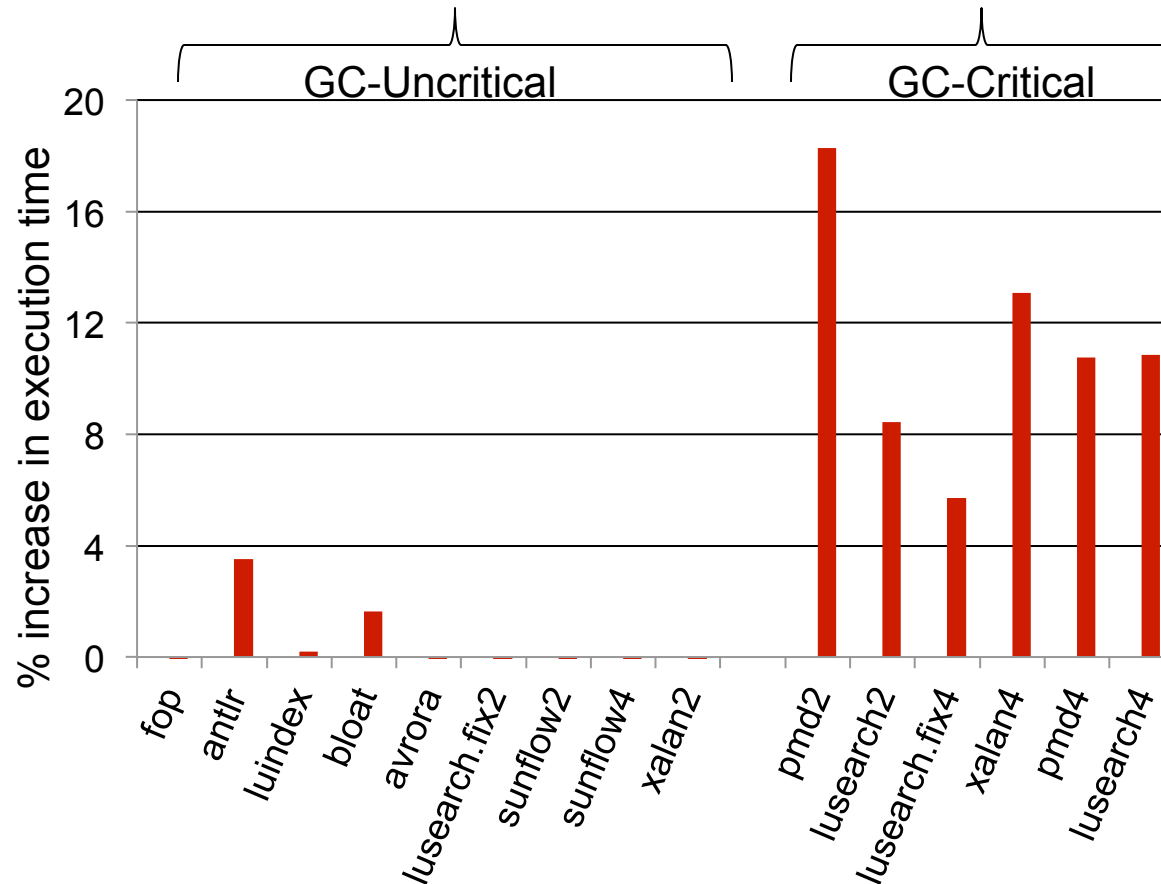


Testing GC's Criticality



Run concurrent collector on LITTLE vs. BIG and measure the difference in execution time

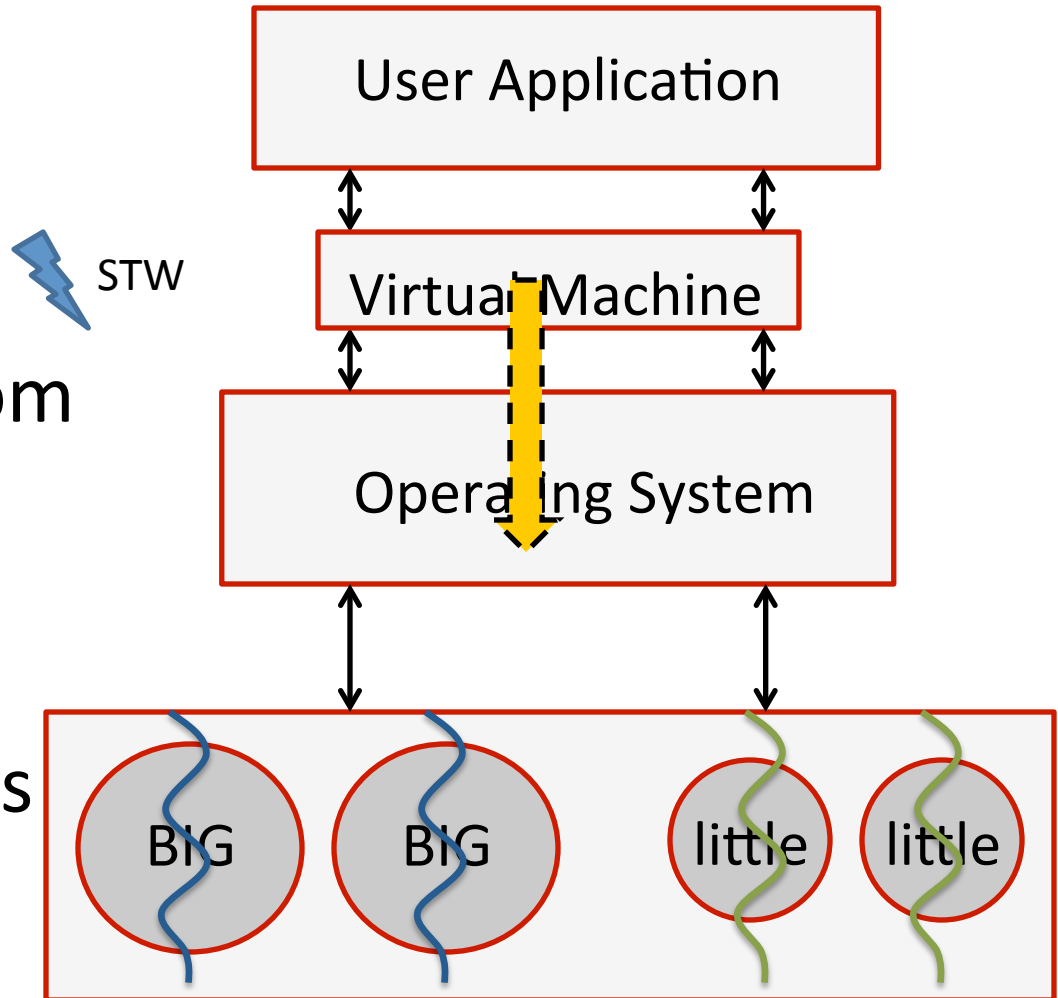
GC Criticality



Run concurrent collector on LITTLE vs. BIG and measure the difference in execution time

Our Adaptive Scheduler

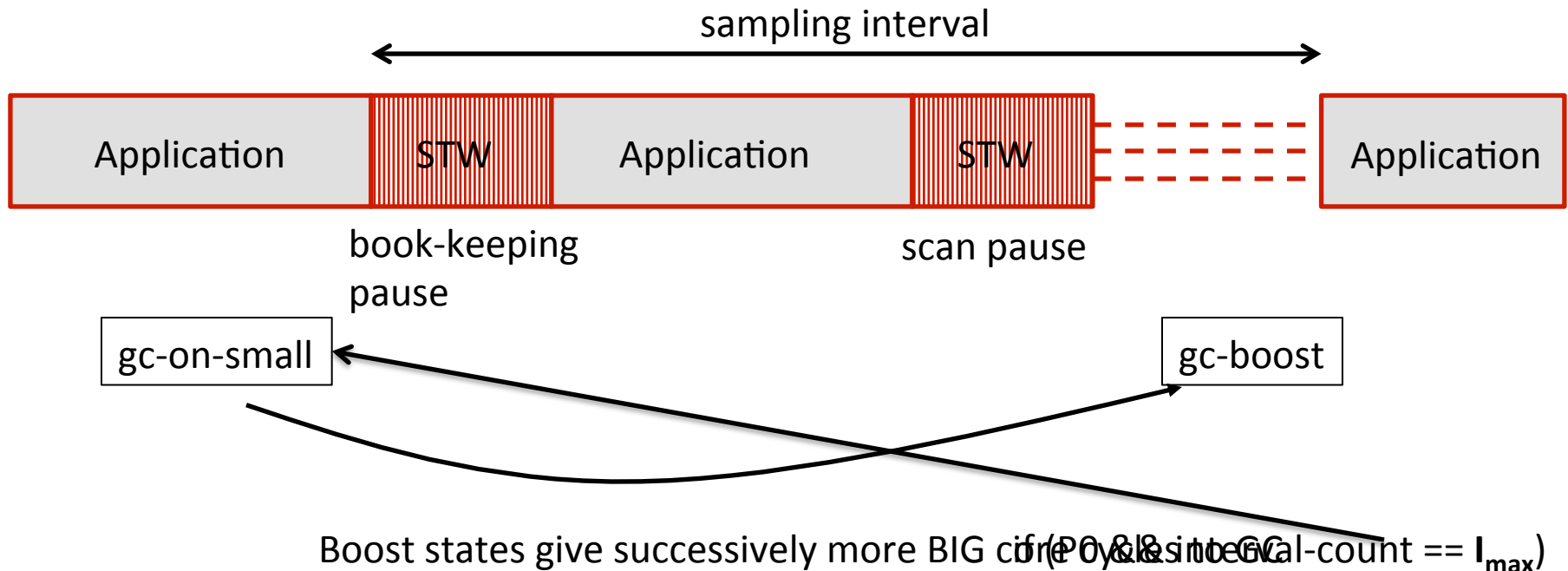
- Measures GC criticality during runtime
- Communicates from JVM down to scheduler
- Scheduler dynamically adapts big core cycles given to GC



Other Schedulers

- Baseline = “gc-on-small”
- gc-fair
 - All threads equal time on the big core(s)
 - Round-robin
 - Based on K. Van Craeynest, S. Akram, W. Heirman, A. Jaleel, and L. Eeckhout. Fairness-aware Scheduling on Single-ISA Heterogeneous Multi-cores. In PACT 2013.

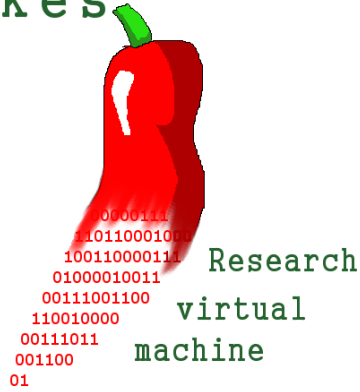
Our GC-criticality-aware Scheduler



State	How many ms scheduled on the BIG core
P0	First GC thread = 1 ms, Second GC thread = 1 ms
P1	First GC thread = 1 ms, Second GC thread = 2 ms
...	

Sampling interval is a variable parameter (different from scheduling quantum)

Jikes



Concurrent garbage collector
2 GC threads



- BIG core is OOO
- Small core is in-order
- 3-level cache hierarchy
- We vary the small core f
- We vary the # BIG cores
- We vary the total # cores

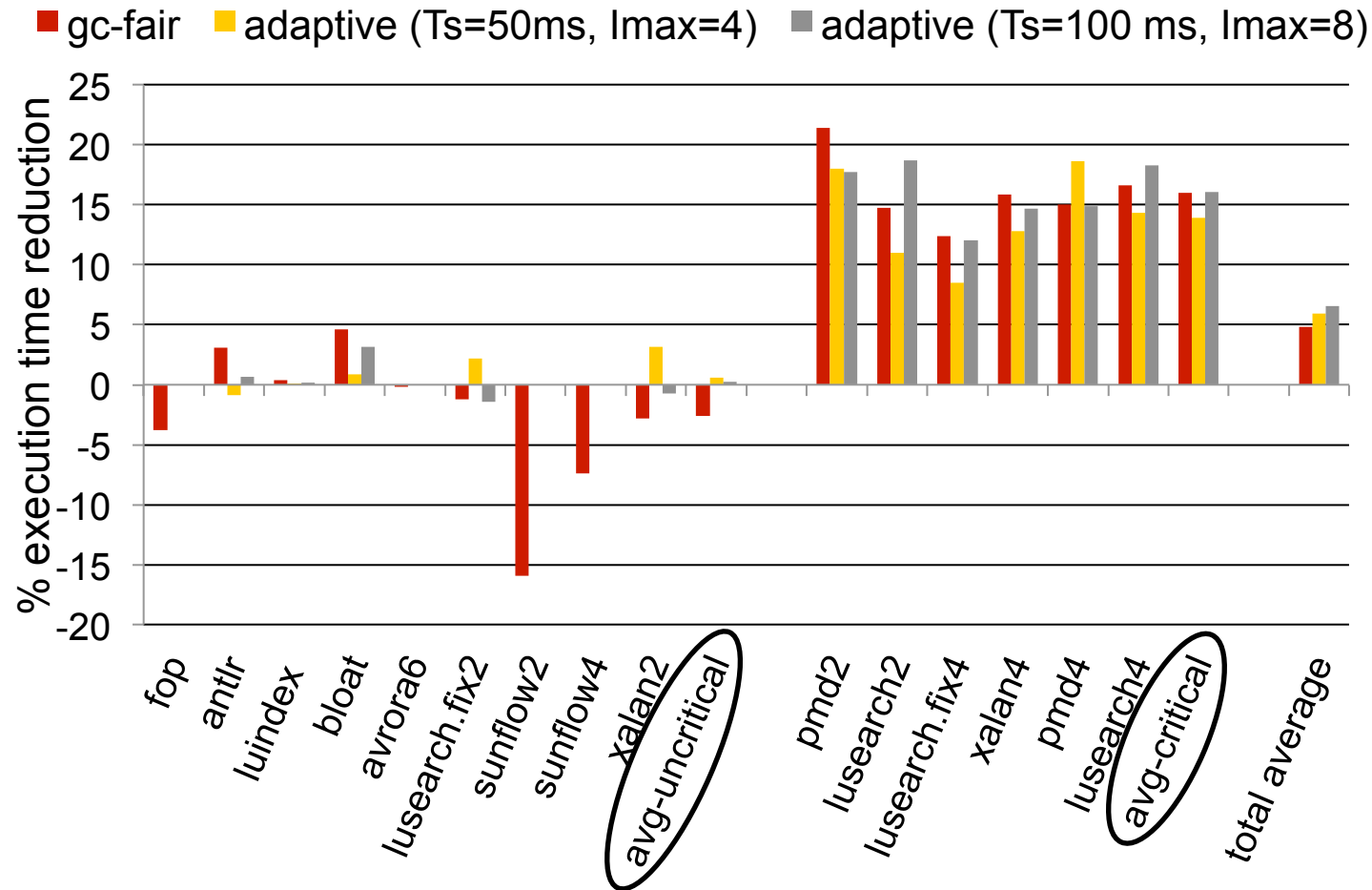


4 single-threaded and 5 multithreaded
We vary the number of threads (2-4)

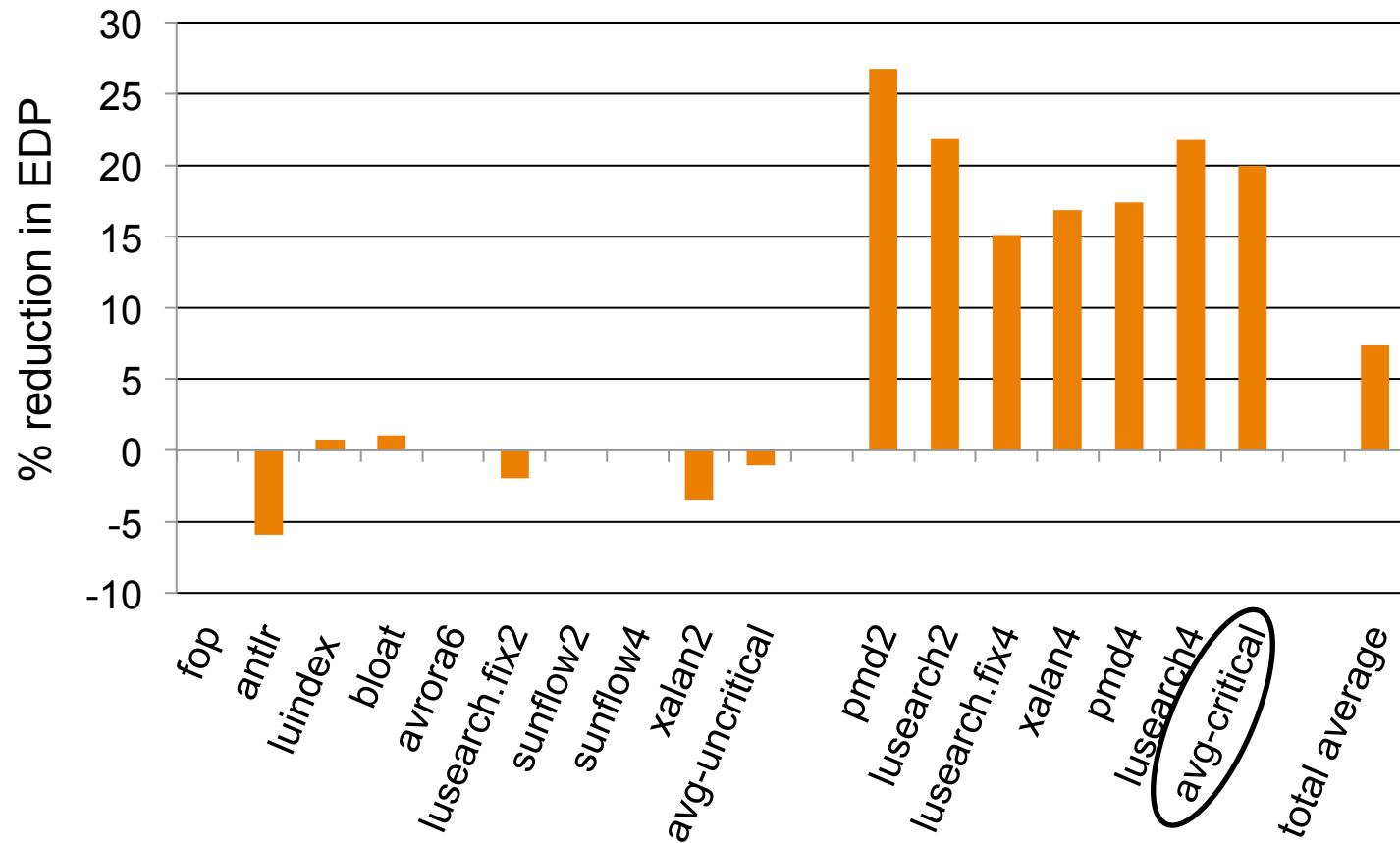
Experimental Results

- Lots of heterogeneous architectures
 - We will show 3B1S
- Baseline = gc-on-small
- Our adaptive scheduler, varying
 - Sampling interval
 - l_{\max}

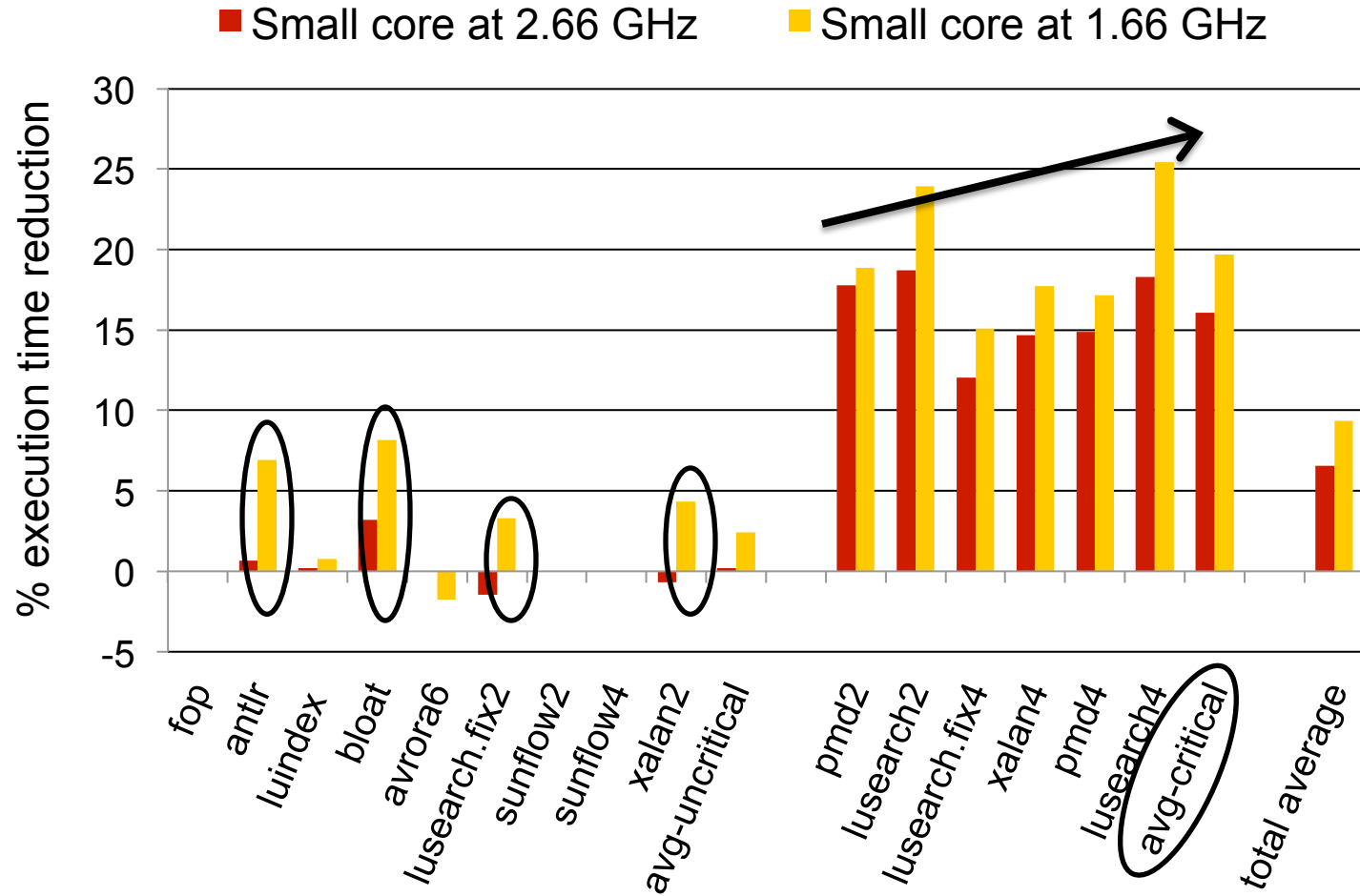
Performance of GC-criticality-aware Scheduler






Energy Efficiency of GC-criticality-aware Scheduler

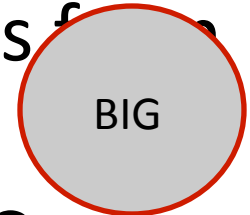


Performance: Small Core Slower



Conclusions

- Concurrent garbage collection benefits from out-of-order execution
- Multi-threaded applications exhibit GC-criticality  STW
- Our GC-criticality-aware scheduler dynamically gives GC more big core time based on information from 
 - Improves performance and energy for GC-critical applications 

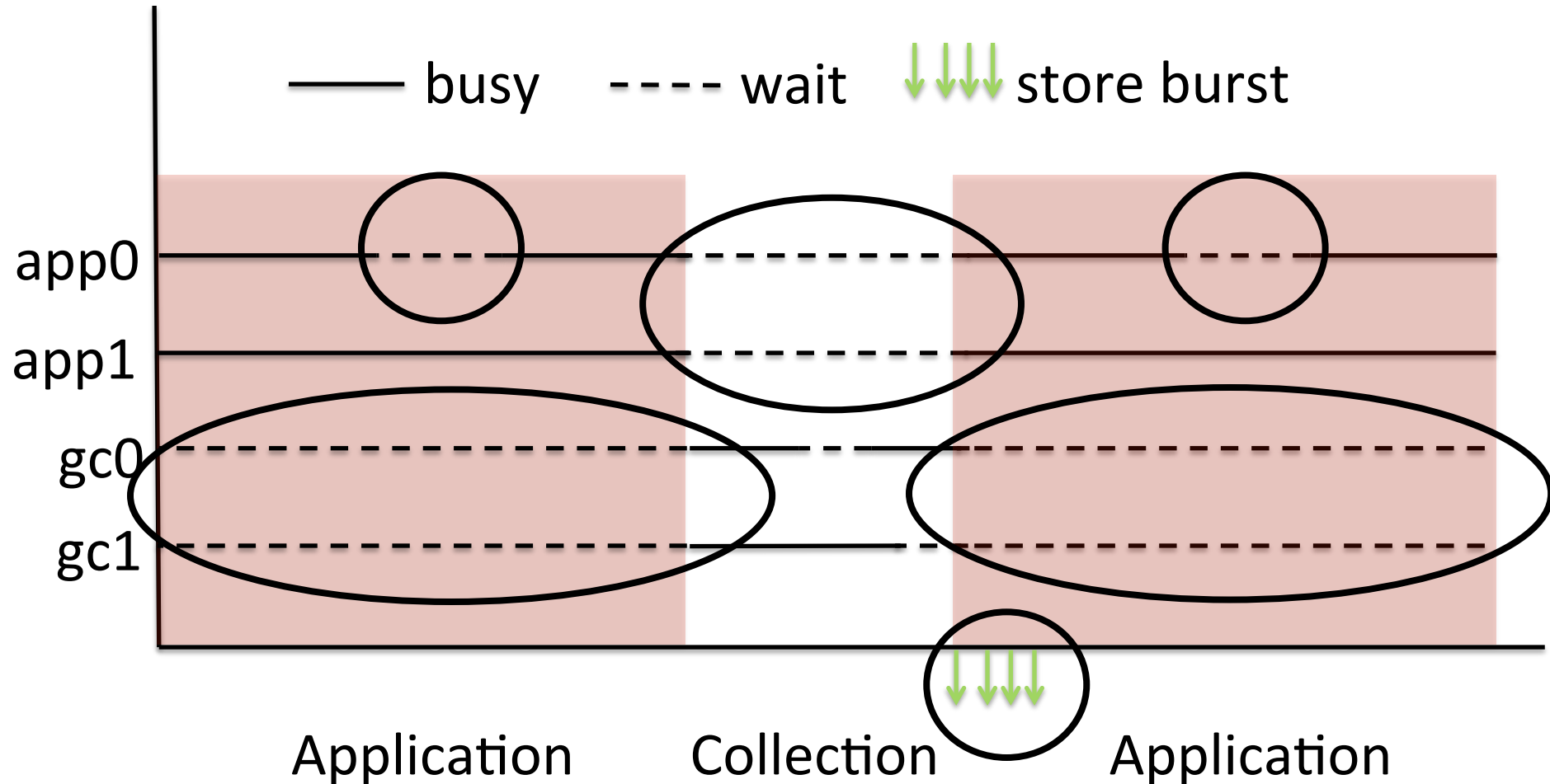


DVFS PERFORMANCE PREDICTION FOR MANAGED MULTITHREADED APPLICATIONS

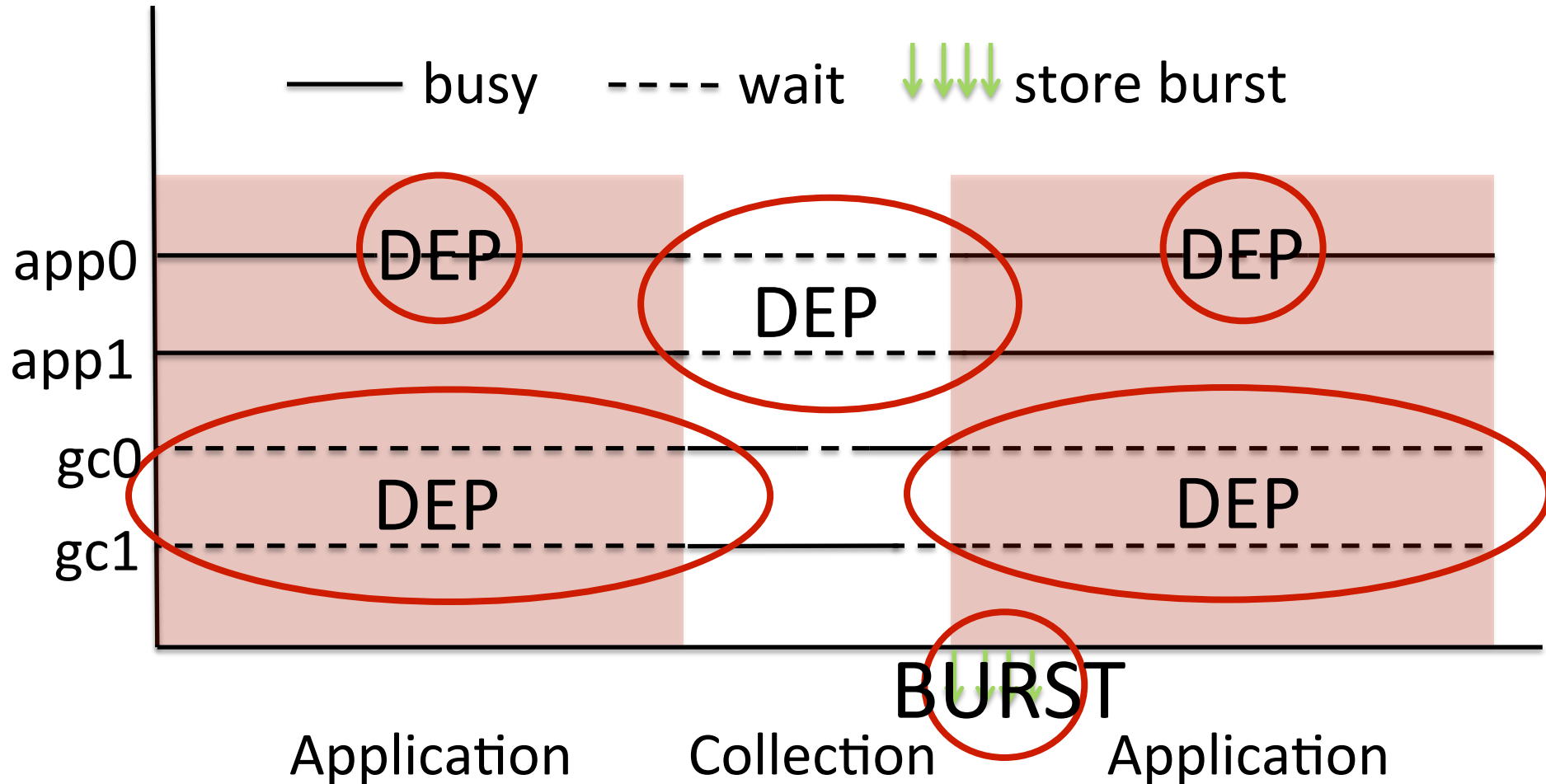
[ISPASS 2016]

Shoaib Akram, **Jennifer B. Sartor**, Lieven Eeckhout
Ghent University, Belgium

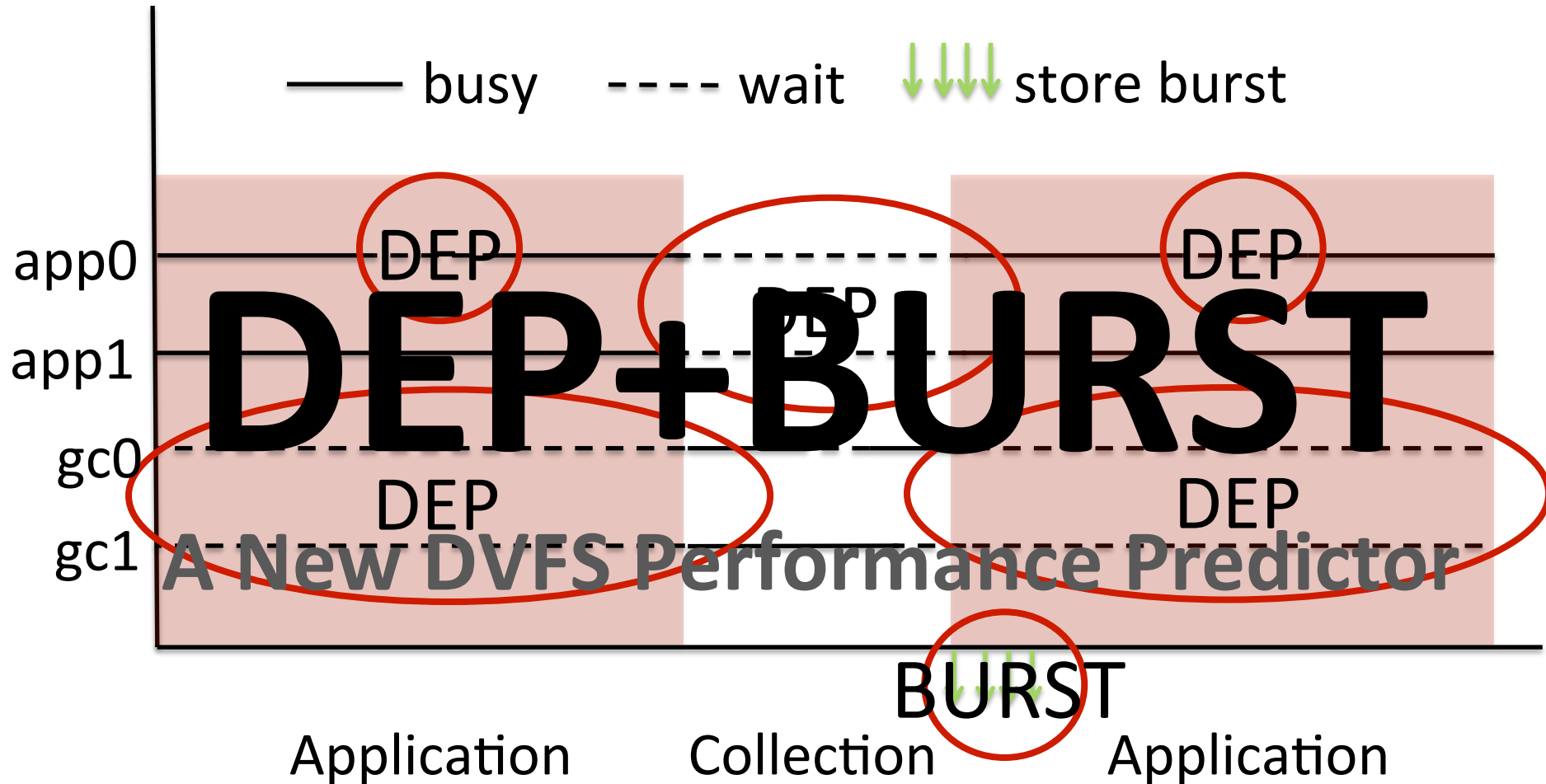
Sources of Inaccuracy in M+CRIT



Sources of Inaccuracy in M+CRIT



Our Contribution

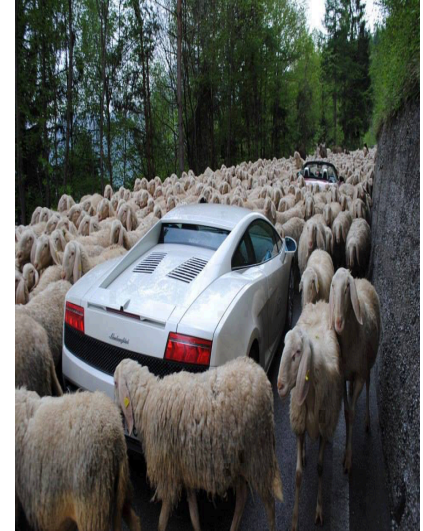


Our Contribution

DEP+BURST

A New DVFS Performance Predictor

Managed Multithreaded Applications



Heterogeneity

Synchronization

Store Bursts

Conclusions

- **DEP+BURST**: First predictor that accounts for
 - Application and service threads
 - Synchronization → inter-thread dependencies
 - Store bursts
- High accuracy
 - Less than 10% estimation error for Java benchmarks
- Negligible hardware cost
- Demonstrated energy savings
 - 20 % avg. for a 10% slowdown (mem-intensive Java apps.)

If You Are Interested

J.B. Sartor, W. Heirman, S.M. Blackburn, L. Eeckhout and K.S. McKinley. **Cooperative Cache Scrubbing**. International Conference on Parallel Architectures and Compilation Techniques (PACT), pages 15-26, Edmonton, Alberta, Canada, August 2014.

S. Akram, J.B. Sartor, K. Van Craeynest, W. Heirman, and L. Eeckhout. **Boosting the Priority of Garbage: Scheduling Collection on Heterogeneous Multicore Processors**. ACM Transactions on Architecture and Code Optimization (TACO), April 2016.

S. Akram, J.B. Sartor, and L. Eeckhout. **DVFS Performance Prediction for Managed Multithreaded Applications**. IEEE Symposium on Performance Analysis of Systems and Software (ISPASS), Uppsala, Sweden, April 2016.

Thank You!

Jennifer B. Sartor

Jennifer.sartor@elis.ugent.be

<http://users.elis.ugent.be/~jsartor/>